PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: PRICE et al.	Attorney Docket No: SUN1P746/P6332
Application No.: 09/930,807	Examiner: VO, Ted T.
Filed: August 15, 2001	Group: 2191
Title: METHOD AND APPARATUS FOR MANAGING DEFUNCT PROCESSES	Confirmation No.: 3576

CERTIFICATE OF FACSIMILE TRANSMISSION
I hereby certify that this correspondence is being transmitted by facsimile to fax number 571-273-8300 of the U.S. Patent and Trademark Office on May 11, 2006.

Signed: Linda L. Pollock
Linda L. Pollock

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22314

Dear Sir:

The Applicants hereby request review of the rejections in the above-identified application. Appeal is proper due to the claims of this application having been twice rejected. Review is requested for the reasons stated in the accompanying Remarks.

This Request is being filed with a Notice of Appeal.

No amendments are being filed with this Request.

Remarks begin on page 2 of this paper.

REMARKS

In the Final Office Action dated January 13, 2006, the Examiner finally rejected claims 1, 4-14, 17 and 20-24 of the present application under 35 U.S.C. § 102(b) as being anticipated by Silberschatz et al., "Operating System Concepts – Module 21: The Unix System", **http://www.cse.ucsc.edu/~sbrandt/courses/Spring01/111/slides/mod21.1.pdf**, 1999 (hereinafter Silberschatz). The Final Office Action set forth new grounds of rejection and therefore the rejections set forth in the preceding Office Actions are not discussed herein.

A request for reconsideration was filed on March 10, 2006, describing some of the differences between the claimed invention and Silberschatz. An Advisory Action was mailed on April 4, 2006, stating that the request had been considered by the Examiner but did not place the application in conditions for allowance, essentially for the same reasons that were stated in the Final Office Action.

The Examiner Mischaracterized the Teachings of Silberschatz

It is respectfully submitted that the Examiner has mischaracterized the teachings of Silberschatz, and inappropriately applied these mischaracterizations to the limitations in the claims of the present application.

Simply stated, the main issue to be appealed is as follows: Silberschatz does neither teach nor suggest (i) the *modification* of the *running* parent process associated with the defunct child process, or (ii) the *creation* of an *agent* thread ***inside*** the running parent process that is operable to *force* the running parent process to collect exit information for the defunct child process, as claimed by the applicant.

Silberschatz is a set of slides providing a general characterization of the Unix system. Numerous topics are described in the slides, such as, Unix's history, design principles, programmer interface, user interface, file system, and so on (*see* slide 21.1 for a complete list of the topics). Slides 21.11 and 21.12 are used by the Examiner in rejecting the claims and describe various Process Control features in Unix. Slide 21.11 explains the concepts of "process", "process identifier", "process control system calls" and "zombie process." Slide 21.12 illustrates how a "fork" system call creates a parent process and a child process, how an "exit" system call terminates the child process, and how a "wait" system call in the parent process provides the process id of the terminated child process, so that the parent can tell which child terminated.

All of these system calls are conventional system calls that are well-known among those of ordinary skill in the art, and are thus discussed in the "Background of the invention" section of the present application. The methodology illustrated in slide 21.12 is described on page 1, lines 9-23, in the applicants' specification as a conventional technique. The specification further lists some drawbacks with this method on page 2, lines 7-14, for example, that the parent process may somehow be flawed and fail to wait for the exit information of its child process, which leads to a defunct child process or so called "zombie process." In this situation, the parent process may have to be terminated, or even the whole computer system may have to be restarted, in order to collect the exit information from the defunct child process, as described on page 2, line 29, through page 3, line 13. Both of these solutions are very disruptive, and thus the applicants' invention is directed to presenting mechanisms for avoiding such disruptions.

As can be seen in claim 1, the invention provides:

> "A computer-implemented method for a running parent process to collect exit information from a defunct child process associated with the running parent process, the computer-implemented method comprising:
>
> identifying a running parent process associated with the defunct child process; and
>
> modifying the running parent process associated with the defunct child process by creating an agent thread inside the running parent process, the agent thread being operable to force the running parent process to collect exit information for the defunct child process, thereby enabling the collection of exit information for the defunct child process associated with the running parent process without terminating the running parent process."

That is, an *agent thread* is created *inside the running parent process*, which *forces* the running parent process to collect exit information for the defunct child process *without terminating* the running parent process. This technique is described in detail in FIG. 6 of the application and on page 16, line 19 through page 18, line 5. As can be seen in this section of the specification, this technique provides a new type of thread that allows modification of the parent process without co-opting any of the target programs' existing thread. The agent thread executes explicit calls to collect exit information, and limits the risk that normal operation of the parent program will be impaired.

In the Final Office Action, the Examiner alleges that slide 21.11, third bullet, fourth and fifth indentations teach the "modifying..." step of claim 1. These passages read as follows:

> *"A parent may wait for a child process to terminate; wait provides the process id of a terminated child so that the parent can tell which child terminated,"*

and

> *"wait3 allows the parent to collect performance statistics about the child."*

With respect to the "...thereby enabling the collection of exit information..." language in claim 1, the Examiner contends on page 3 of the Final Office Action that

> *"process control shown in slides, 21.11 and 21.12 will cause the shell process of slide 21.12 running (Refer to the fourth bullet in the slide 21.11)"*

This is clearly not the same as creating an agent thread inside the running parent process, which forces the running parent process to collect exit information for the defunct child process without terminating the running parent process, as described in claim 1. Without going into any further details of the invention, it ought to be clear that there is no way that these sections of Silberschatz anticipate or render obvious the claimed methods, and that the Examiner appears to have misunderstood the present invention and how it relates to the teachings of Silberschatz.

All the independent claims in the application have limitations that include the creation of an agent thread inside the running parent process, similar to what has been discussed above with respect to claim 1. Thus, in view of the foregoing, it is respectfully submitted that all pending claims are patentably distinct over the art of record and that the outstanding rejections be withdrawn. Should the conferees have any questions or concerns regarding the present application, they are encouraged to contact the undersigned at the telephone number set out below.

Respectfully Submitted,
BEYER WEAVER & THOMAS, LLP

Fredrik Mollborn

Fredrik Mollborn
Reg. No. 48,587

P.O. Box 70250
Oakland, CA 94612-0250
(650) 961-8300